Yaw control using rear wheel steering

Examensarbete utfört i Reglerteknik
vid Tekniska Högskolan i Linköping
av

Daniel Westbom
Petter Frejinger

Reg nr: LiTH-ISY-EX-3273
Yaw control using rear wheel steering

Examensarbete utfört i Reglerteknik
vid Tekniska Högskolan i Linköping
av

Daniel Westbom
Petter Frejinger

Reg nr: LiTH-ISY-EX-3273

Supervisor: Sigvard Zetterström
            Claes Olsson
            Niclas Persson

Examiner:  Svante Gunnarsson

Linköping, 21st November 2002.
Abstract

The purpose of this project is to continue the work on a vehicle model developed in ADAMS/Car and applied with the concept of ACM (Autonomous Corner Module). The project is divided up in two parts. The objective of the first part is to setup a co-simulation environment between ADAMS/Car and MATLAB/Simulink, and evaluate the vehicle model. In the second part a yaw controller is developed using only the rear wheel steering possibilities. The controller will be evaluated when it is applied on the vehicle model.

The approach is to develop two models, one simpler in MATLAB/Simulink and one more complex in ADAMS/Car, and verify that they show similar behavior. The models will then be linearized and the control design will be based on the most appropriate linear model. Most of the work has been developing and evaluating the two vehicle models in ADAMS/Car and MATLAB/Simulink.

The result was a working co-simulation environment where an evaluation of two different controllers was made. Due to linearization of the ADAMS model was unsuccessful, the controllers were based on the simpler linear Simulink model. Both controllers show similar results. Tests on the ADAMS model showed that it is hard to control both the yaw rate and body slip only by rear wheel steering.
Acknowledgement

First of all we would like to thank Volvo Cars for offering us this project. Furthermore we want to thank Ahmed El-bahrawy and supervisors Claes Olsson and Sigvard Zetterström for supporting us at Volvo. The support at MDI would we also like to thank for helping us with the ADAMS environment. Finally we want to thank our examiner Svante Gunnarsson and supervisor Niclas Persson from the University of Linköping.
Notation

Symbols

$\Delta \beta$ Body slip error.
$\Delta \dot{\Psi}$ Yaw rate error.
$\Delta_G$ Model error.
$\Phi, \Theta, \Psi$ Roll-, Pitch and Yaw angle of the vehicle.
$\alpha_i$ Slip angle.
$\beta$ Body slip.
$\delta_{ack}$ Ackermann angle.
$\delta_f$ Front wheels’ steering angle.
$\delta_i$ Steering angle each wheel.
$\delta_r$ Rear wheels’ steering angle.
$\eta$ Constant.
$\dot{\Psi}$ Yaw rate.
$\omega$ Rotation vector
$A$ State-space matrix A.
$B$ State-space matrix B.
$C$ State-space matrix C.
$D$ State-space matrix D.
$F$ Force.
$J$ Moments of inertia.
$K$ Kalman filter gain.
$L$ Linear quadratic gain.
$M$ Momentum.
$N$ State-space disturbance matrix N.
$Q$ Penalty matrices.
$R$ Rotation matrices.
$S$ Sensitivity function.
$T$ Complementary sensitivity function.
$V$ Symbolize any vector.
$X, Y, Z$ Global axis coordinates.
$a$ Distance center of gravity to front axle (along x-axis).
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>Distance center of gravity to rear axle (along $x$-axis).</td>
</tr>
<tr>
<td>c</td>
<td>Distance center of gravity to wheel (along $y$-axis).</td>
</tr>
<tr>
<td>d</td>
<td>Damping constant.</td>
</tr>
<tr>
<td>h</td>
<td>Distance center of gravity to road.</td>
</tr>
<tr>
<td>k</td>
<td>Stiffness constant.</td>
</tr>
<tr>
<td>m</td>
<td>Mass of the vehicle.</td>
</tr>
<tr>
<td>n</td>
<td>Measurement noise.</td>
</tr>
<tr>
<td>u</td>
<td>Input signal.</td>
</tr>
<tr>
<td>v</td>
<td>Velocity center of gravity.</td>
</tr>
<tr>
<td>$v_i$</td>
<td>Translational velocity on wheel $i$.</td>
</tr>
<tr>
<td>w</td>
<td>Disturbance.</td>
</tr>
<tr>
<td>x</td>
<td>State vector.</td>
</tr>
<tr>
<td>$x, y, z$</td>
<td>Local axis coordinates.</td>
</tr>
<tr>
<td>$\hat{x}$</td>
<td>Estimated state vector.</td>
</tr>
<tr>
<td>y</td>
<td>Output signal.</td>
</tr>
<tr>
<td>$i$</td>
<td>Index wheel.</td>
</tr>
<tr>
<td>$N$</td>
<td>Index normal force.</td>
</tr>
<tr>
<td>c</td>
<td>Index lateral force.</td>
</tr>
<tr>
<td>1</td>
<td>Index left front wheel.</td>
</tr>
<tr>
<td>2</td>
<td>Index right front wheel.</td>
</tr>
<tr>
<td>3</td>
<td>Index left rear wheel.</td>
</tr>
<tr>
<td>4</td>
<td>Index right rear wheel.</td>
</tr>
</tbody>
</table>

**Abbreviations**

- **ACM**: Autonomous Corner Module
- **SIMO**: Single Input Multiple Output
- **LQ**: Linear Quadratic
- **ADAMS**: Automatic Dynamic Analysis of Mechanical Systems
- **MATLAB**: MATrix LABoratory
- **MDI**: Mechanical Dynamics Inc
- **ADAMS model**: The vehicle model developed in the ADAMS environment
- **Simulink model**: The vehicle model developed in the Simulink environment
# Contents

1 Introduction .................................................. 1  
  1.1 Background ................................................. 1  
  1.2 Purpose .................................................... 1  
  1.3 Statement of problem ...................................... 1  
  1.4 Approach .................................................. 2  

2 Autonomous Corner Module ................................. 3  
  2.1 ACM applications .......................................... 5  
  2.2 Conclusion ................................................ 6  

3 Environment .................................................. 7  
  3.1 ADAMS/Car ................................................. 7  
  3.2 MATLAB/Simulink ......................................... 8  
  3.3 Co-simulation ............................................... 8  

4 ADAMS model ................................................ 11  
  4.1 Model description .......................................... 11  
  4.2 Input and output signals .................................. 15  
  4.3 Co-simulation ............................................... 15  
  4.4 Linear ADAMS model ....................................... 16  
    4.4.1 Ways to avoid linearization problem ............... 16  
    4.4.2 Other modelling issues ............................... 18  

5 Simulink model ............................................... 23  
  5.1 Model overview .......................................... 23  
  5.2 Coordinate systems ....................................... 24  
  5.3 Forces and torques ....................................... 24  
  5.4 Tire model ................................................. 26  
  5.5 Suspension model ......................................... 28  
  5.6 Dynamics .................................................. 29  
  5.7 Global positioning ....................................... 30  
  5.8 Linear Simulink model .................................... 30  

11
6 Model evaluation 33
  6.1 Driving scenario .............................. 33
  6.2 Tuning process ................................ 35
  6.3 Non linear model comparison ....................... 36
     6.3.1 Speed and movements .......................... 36
     6.3.2 Forces ........................................ 37
  6.4 Linear model evaluation .......................... 38
  6.5 Conclusion ...................................... 38

7 Active yaw control 41
  7.1 Control strategy .................................. 42
     7.1.1 Error system .................................. 42
     7.1.2 Signal separation .............................. 43
  7.2 Proportional feedback .............................. 44
     7.2.1 P-controller characteristics ..................... 44
  7.3 Linear quadratic control .......................... 46
     7.3.1 Extended error system ......................... 46
     7.3.2 LQG controller ................................ 46
     7.3.3 LQG controller characteristics ................ 48
  7.4 Evaluation of the controllers ....................... 50
     7.4.1 Evaluation using Simulink model simulations .. 51
     7.4.2 Evaluation using ADAMS model simulations .... 52
     7.4.3 Conclusion .................................... 53

8 Conclusion 61
  8.1 Autonomous Corner Module .......................... 61
  8.2 The ADAMS and Simulink model ...................... 61
  8.3 Control design .................................... 62

A Two mass system 67
Chapter 1

Introduction

This Master of Science thesis was carried out at Volvo cars in Gothenburg. Volvo cars is a world wide company that since 1999 is a part of the Ford Motor Company, which is the world’s second largest car manufacturer. The brand Volvo is associated with the properties safety, quality and environment.

1.1 Background

The Autonomous Corner Module (ACM) was patented 1998 at Volvo cars, [14], and is a wheel corner of a vehicle that is individually powered and has computer controlled propulsion, braking, steering, spring and damper functions. The idea is the foundation for cooperation between Volvo cars and Linköping University, called Evolve. The project is planned for three years, starting 2001, and the goal is to implement the wheel corners on a Volvo V70. Parallel with this project an ADAMS/Car vehicle model was created in a previous thesis work, [7], and applied with ACM. The model is the corner stone in this project.

1.2 Purpose

The purpose of this project is to continue the work on a vehicle model applied with the concept of ACM in an ADAMS/Car environment. By using the model, dynamic behavior of the vehicle can be studied as well as different control systems.

1.3 Statement of problem

The objective of the project can be divided into three different parts, where the first is to correct model errors in the existing ADAMS/Car model. The second part is to set-up a co-simulation environment between ADAMS/Car and MATLAB/Simulink, and evaluate the model. The development of a yaw controller by
using rear wheel steering, and evaluation on the ADAMS/Car model, is the third and last part.

1.4 Approach

The traditional way of developing control systems is by using models, implemented in MATLAB/Simulink, that represent a physical system. Often the dynamic behavior of the physical system is approximated and therefore the model is characterized by simplifications. The result of a controller applied on the physical system depends on how well the model describes the system. By the use of ADAMS/Car, modelling complex system is possible and consequently the possibility of a successful controller increase. The two ideas, controller based on simple and complex model respectively, do not necessarily serve as the contrast to each other. A simple model would definitely be preferable if it includes all important dynamic behavior. A comparison between the complex and the simple model tells if this is true. In this project the two ideas are developed parallel with each other. The existing ADAMS/Car model serves as the complex model, while a simplified model is created and implemented in Simulink. Based on the fact how well the Simulink model turns out, a decision is made on which model to use when developing the yaw rate controller. The objective is to control the yaw rate of the non-linear ADAMS/Car model.

The project is structured in the following way to solve the three problems presented in Section 1.3. First of all the model in ADAMS/Car must be considered okay, and therefore necessary modifications is the main priority. In ADAMS/Car predefined driving scenarios are provided for simulation, which force the vehicle to behave a certain way. However, to fulfill the objectives in this project optional driving scenarios must be possible to simulate, and thus a co-simulation between MATLAB/Simulink is convenient, see Chapters 3, 4, and 6. A working co-simulation environment sets the conditions for implementing a latter controller. To compare the courses of action of deriving yaw controllers by a complex and a simple model respectively, a simple model needs to be implemented in Simulink, see Chapter 5. It should have the possibility to steer each wheel individually in line with the concept of ACM. A comparison between non linear and linear models will tell how well they resemble each other, see Chapter 6. Based on the comparison of the result, one of the models is used in the work with deriving a yaw rate controller, see chapter 7. The objective of the project is fulfilled when the controller is applied and evaluated on the non-linear ADAMS/Car model.
Chapter 2

Autonomous Corner Module

The Autonomous Corner Module (ACM) was patented 1998 and is a wheel corner that is individually powered and has computer controlled propulsion, braking, steering, spring and damper functions, [14]. The unit is supposed to be identical on the left and right front axle as well as the left and right rear axle. It has two mechanical interfaces with the body of the vehicle via the suspension arms. The rest is based on the idea of x-by-wire\(^1\) technique.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{autonomous_corner_module}
\caption{Overview of an Autonomous Corner Module, where (1) is steering actuators and (2) rotational spring-damper system}
\end{figure}

In Figure 2.1 a wheel corner in accordance with the concept of ACM is illustrated. It is supplied with two actuators (1) for steering and a rotational spring-damper system.

\(^1\)X-by-wire is an expression for electrically controlled mechanical parts.
damper system (2). The steering actuators were originally thought of as electromechanical systems, e.g. ball roller screw, but can also be implemented electro hydraulically. The latter is used in the Evolve project. The damper is supposed to be based on the concept of magnetorheological fluid. The fluid is trapped in a rotating housing and is continuously affected by the magnetic field of two electric solenoids. Changes in the intensity of the field change the viscosity of the fluid and give the damper appropriate characteristics. The spring system is divided in a passive and active part. The first can be a leaf spring with the purpose of taking care of the static load of the vehicle. The active part of the spring system is a torsion spring that consists of eight identical rubber torsion discs, which is more or less pre-tension by a motor depending on the driving situation. This solution of the spring-damper-system creates conditions for the suspension to be actively controlled. For the braking and the propulsion of the vehicle, each wheel is provided with a brake system and an electric motor.

Building a car based on the concept of x-by-wire and ACM enables lots of new possibilities in different perspectives. Jerome Jimenez, [7], summarized it in points of safety, environmental and additional features.

**ACM safety aspects**

- Absence of drive shafts facilitates safe power plant packaging anywhere in the vehicle.
- Absence of accelerator reduces the risk of foot injuries in crash situations.
- Absence of steering column enables enhanced driver protection.
- Pedals need no connection with crash critical components and they are thus easy to make adjustable.
- Double steering actuators per wheel enable the possibility to steer even if one actuator should fail.
- Individual torque and steering on each wheel makes directional control possible even if one wheel is failing completely.
- Three separate brake systems, i.e. wheel motor, friction brake and maximum toe angle.
- Appropriate dynamic behavior can be fulfilled by using a combination of different active control systems.

**ACM environmental aspects**

- A weight reduction compared with conventional chassis and power train systems.
- Fluids in brake and steering systems are removed.
2.1 ACM applications

- A number of complex components and subsystems are deleted and replaced by a large number of identical components. This helps production optimization and minimizes storage and transport.
- Aluminium components and electrical cables can use recycled material.
- Dynamically optimized wheel angles reduce tyre wear.
- Regenerated brake energy fed into batteries and capacitors reduces fuel consumption and brake lining wear.

ACM additional features

Software’s characteristic that are available for customization:

- Personal ride comfort can be chosen by adjustments to initial settings of the wheel corners.
- Vehicle response in terms of over- and under steer behavior may be tuned within limits.
- Damper mode settings such as sport, comfort and racing.
- Parking aid steer, e.g. diagonal motion.
- Built-in jack using spring force control.

2.1 ACM applications

Safety was, from the beginning, the main purpose of implementing control systems in vehicles. In the late seventies anti-lock braking systems were introduced followed by traction control in the mid eighties. In the nineties the stability controller was developed and now, in the beginning of the twenty-first century, more sophisticated control systems are developed. Active steering systems, which are enabled by ACM, make it possible to steer each wheel individually. It can be helpful for assisting the driver in different driving situations. At lower velocities, as parking or tight cornering, the rear wheels can help the driver decreasing the turning radius. At higher velocities it might be desirable to have some stabilizing assistance by, for example, an active yaw control using rear wheel steering. Another interesting point is how to decide when the car should behave in a certain way.

Active suspension

An active suspension system has the capability to adjust itself continuously to changing road conditions. It extends the design parameters of the system by constantly monitoring and adjusting itself, thereby changing its characteristics on an ongoing basis. Consequently, active suspension offers good handling, responsiveness and safety, [11]. The design of the spring damper system in the Autonomous
Corner Module offers the possibility to use active suspension in the vehicle, as mentioned in the beginning of the chapter. Some passenger vehicles offer fully active suspension systems as standard or optional equipment. Some examples are Cadillac, Land Rover and Mercedes-Benz.

**Active yaw control**

The active yaw control system is supposed to stabilize the vehicle by applying a yaw-torque. Many of the major vehicle manufacturers have tested such a system and some of them have started to produce models with the system included. The active yaw control is effectively used in high-speed corner situations, where the vehicle will appear more stable, [3]. The vehicle will also appear significantly safer and gentler during abrupt braking manoeuvres. Most of today’s systems distribute optimal braking power on each wheel when instability is detected, both in high speed cornering and braking manoeuvres as well as small slip and steering angle instabilities. The use of ACM offers, for example, the possibility to steer the rear wheels to generate a compensating yaw motion.

**Side wind- and road-slope compensation**

Another interesting idea to study, with the concept of ACM, is a side wind compensation system to stabilize the vehicle. When the vehicle travels straight and experiences a sudden or constant wind, the control system compensates and gives the driver a normal and stable feeling of the car. The same idea concerns a vehicle travelling on a sloped surface. Forces acting on the car would give the driver a different feeling than normal conditions, but with a road slope compensating control system the driver will get the same feeling as if he or she is travelling on a flat surface. With the concept of steer-by-wire the idea of side wind- and road slope compensation has been brought up. However, these ideas are still on a research and test stage.

**2.2 Conclusion**

The concept of ACM enables a lot of new control systems from what some have been presented in this chapter. Based on the sections above, an active yaw controller was chosen to be studied further. The reason was that it would be interesting to see how a compensating torque could be generated without using the brakes, and that the system is simple to implement. The following chapters will describe the vehicle models and an active yaw controller using the possibility to steer the rear wheels.
Chapter 3

Environment

3.1 ADAMS/Car

ADAMS, [1], is the name of a group of products whose purpose are mechanical modelling in different areas. In the vehicle line of business the program is called ADAMS/Car, and it is provided with typical car components and simulation situations. Each ADAMS program can be combined with useful modules, which will be described in Section 3.3.

ADAMS/Car is, as mentioned, a specialized environment for modelling of vehicles. It allows the user to create virtual prototypes of vehicles, and analyze them much like physical prototypes would be analyzed. By the use of advanced models, ADAMS/Car makes it possible to study complex dynamic behavior. The ADAMS/Car model hierarchy consists of templates, subsystems and assemblies. Templates include all necessary geometry, constraints, forces, and measurements. Subsystems are based on templates and allow the user to change the parametric data of the template. One or several subsystems can be grouped together to form assemblies. Standard assemblies in ADAMS/Car are e.g. suspension assembly, full-vehicle assembly. The hierarchy of the ADAMS/Car model used in this work is illustrated in Figure 4.2.

By the use of ADAMS/Car it is possible to evaluate the performance of a design and make modifications before building and testing a physical prototype. Design changes can be analyzed faster and at lower cost than physical prototype testing would require.

Models in the ADAMS/Car environment can be linearized by the use of the module ADAMS/Linear. There are two different types of linearization procedures. The first one generates eigenvalues and modes of the model, which can be imported to ADAMS/Car and animated via a graphical interface. The second linearization process creates a states space model that can be imported into the MATLAB/Simulink environment for further analysis and simulations.
3.2 MATLAB/Simulink

Simulink, [10], is a tool that gives the user the ability to model and analyze dynamic behavior of mathematical models via a graphical interface, see Figure 3.1. It is completely integrated in the computer program MATLAB, [9], which serves as a mathematical tool for engineers all over the world. MATLAB contains several analysis tools that make the combination with Simulink useful not only in control system design but also in other simulation applications. In this project linear models will play an important role. Simulink includes linearization functions for its models, which make the process easier. All the plots included in this report have been created in the MATLAB environment.

![Figure 3.1. The graphical environment of Simulink](image)

3.3 Co-simulation

By combining ADAMS/Car and MATLAB/Simulink it is possible to, for example, add control algorithms to a model developed in ADAMS/Car. The synchronized simulation of the two systems is called a co-simulation. In order to set up a co-simulation environment ADAMS/Car has to be provided with two modules, ADAMS/Controls and ADAMS/Solver. The ADAMS/Controls module generates a simulation model based on the ADAMS/Car model, which can be imported into Simulink. ADAMS/Solver’s purpose is to calculate the result from the equations of motion. During a co-simulation, a closed loop between the ADAMS/Car model and the control system is formed, see Figure 3.2. ADAMS/Car inputs of a model enter the ADAMS/Solver, which calculates the output signals from the model. The ADAMS/Solver output signals enter the control system, where MATLAB calculates the control signals, and a new iteration starts by sending the control signals as inputs to the ADAMS/Car model. The ADAMS/Solver module is a numerical analysis application that automatically solves the equations of motion for kinematics, static and dynamic simulations for an ADAMS/Car model. The result of the co-simulation can be imported into ADAMS/Car, where plots and animations from the simulation are available.
Figure 3.2. The principle of co-simulation between ADAMS/Car and MATLAB
Chapter 4

ADAMS model

This chapter will describe a vehicle model that is developed in ADAMS/Car, and therefore named the ADAMS model. The model will serve as a foundation for evaluating developed control algorithms and was originally created in a previous work, [7], although not completely evaluated. The following sections will describe the structure of the ADAMS model, while the corrections and tuning work will be discussed in Chapter 6. A section with a brief description of how the model is set up in a co-simulation environment is also provided.

4.1 Model description

The ADAMS model is supposed to resemble a complete vehicle using the ACM, see Chapter 2. Since there is not a real vehicle using the ACM concept today the model can later be adjusted to describe the reality well. Figure 4.1 illustrates the model’s front and rear axle (1), front and rear wheels (2), body (3) and steering column (4).

In ADAMS/Car the model is built as a full vehicle assembly\(^1\) and based on six subsystems, see Figure 4.2. The subsystems are based on one or more templates and are connected via communicators, which are components for exchanging information between subsystems. Each of the components in a template needs one or more hardpoints to define its position. When a subsystem is created each hardpoint’s position can be changed, and thus the components will move with the hardpoints. To simulate the model in a co-simulation environment, necessary input and output signals are defined in the templates. The following sections describe each template of the ADAMS model.

\(^1\)In ADAMS there are either a generic or a full vehicle assembly.
Figure 4.1. Full vehicle assembly of the ADAMS model

Figure 4.2. The ADAMS model with subsystems and templates in a tree structure
Axle template

Figures 4.3 and 4.4 illustrate the flexible body (1), the spring damper systems (2) and the steering actuators (3) in the axle template. The flexible body’s purpose is to hold up the static weight of the vehicle. If the vehicle for some reason is heavily loaded or is experiencing other forces, the spring damper system is supposed to be activated. To make the four-wheel steering possible, a spring and a damper are modelled between two cylinders to resemble a steering actuator (3). Moving the two steering actuators of the wheel corner in opposite directions set the steering angle.

Figure 4.3. The axle template of the ADAMS model

The input signals to the axle template are the forces in the four steering actuators. No output signals are defined for the template.
Wheel template

The wheel template (2), see Figure 4.1, is based on a MF-tire\(^2\), [2]. The coefficients in the tire model are derived from measured data collected from a physical tire, which makes the model realistic. Since the parameters are defined in a property file, it is possible to make some simplifications that are useful in the tuning work in Section 6.2. The Simulink model, derived in the next chapter, is only experiencing lateral tire forces, see Section 5.4. To make the ADAMS model correspond with the Simulink model a tire model is used where only lateral forces act on the tire.

No input signals are defined in the wheel template. The output signals are the left and right global steering angles of the wheel, i.e. the angle difference between an earth-fixed coordinate system and the pointing direction of the wheel.

Body template

The body template (3), illustrated in Figure 4.1, can be used to vary weight and placement of passengers, luggage, fuel and body mass centre. It is possible to adjust these placements by an automatic mass adjustment function in ADAMS/Car, which makes different types of predefined load cases easy to simulate.

There are no input signals to the body template. The output signals are the global Ψ angle in the centre of gravity and the angular velocities \( \Phi, \Theta, \Psi \), see Section 5.2. Furthermore, the local velocities \( v_x, v_y, v_z \) are also output signals. Local velocities mean the velocities a driver experience with respect to the centre of gravity.

\(^2\)Magic Formula tire
Steering template

The steering template is used as a visual aid and due to the fact that a full vehicle assembly require this subsystem. The template does not affect other subsystems in the ADAMS model.

4.2 Input and output signals

To sum up the previous section the input signals to the model are eight cylinder forces. Output signals are $\Psi$, $v_x$, $v_y$, $v_z$, $\dot{\Phi}$, $\dot{\Theta}$, $\dot{\Psi}$ and four global steering angles. The local steering angles $\delta_i$ are calculated by subtracting the $\Psi$ angle from the global steering angles, which is done in the Simulink environment. The body slip, $\beta$, is the angle between the vehicle’s pointing direction and travel direction, which will be of importance later in this report.

4.3 Co-simulation

A co-simulation environment in MATLAB/Simulink has been set-up to simulate the ADAMS model, see Figure 4.5.

![Figure 4.5. Block diagram of the co-simulation environment for the ADAMS model](image)

The block diagram shows input and output signals from the ADAMS model block, generated by ADAMS/Controls, see Section 3.3. Since it is hard to apply a force that gives a specific steering angle, PI controllers are used to set the desired angle. There are eight input signals that represent the forces in the cylinders, since e.g. it may be interesting to include camber angles in the future.
4.4 Linear ADAMS model

ADAMS/Car can be combined with the ADAMS/Linear module that provides functions for linearizing models. Linearization can be done in two different ways, one generating the eigenmodes of the model and the other exporting a linear state space model. The linear model’s eigenvalues are interesting to study in order to get insight into the dynamical behavior of a vehicle. A linear model is stable if the real part of the eigenvalues is negative and unstable if some eigenvalue is positive. Excitation of an unstable mode would result in an unbounded response from the model. To verify how excitation of a mode would affect the system, it is possible to animate each mode in ADAMS/Car. The reason for linearizing the model into a state space form is the possibility to import it into a mathematical analysis environment, such as MATLAB, and to use it in e.g. a control design.

Figure 4.6. The process of linearizing the ADAMS model

The idea is to linearize the ADAMS model by ADAMS/Linear, see Figure 4.6. The input signals are the steering angles and the output signal is the yaw rate. The operating point is 90 km/h driving straight ahead. Unfortunately, the linearization process of the ADAMS model give unstable modes. After discussion with the MDI\textsuperscript{3} support, the instability was explained by the tire model. The linearization process in ADAMS/Linear is based on the assumption that internal forces are determined in the presence of stiffness and damping. The tire forces are functions of slip angles, which are based on the velocity and rotation of the vehicle, see Section 5.4. This absence of stiffness and damping factors in the tire model causes the problem which leads to that the linearized model is not reliable.

4.4.1 Ways to avoid linearization problem

To solve the stability problem described in the previous section, the tire model can be excluded from the ADAMS model and implemented in Simulink instead. A co-simulation will in that case indicate that the system has similar behavior. The new model is called the ADAMS external tires. The procedure, illustrated in Figure 4.7, is derived to achieve a linear ADAMS model.

A generic assembly is used for deriving the ADAMS external tires model. Instead of tires the vehicle is placed on bushings, which is a component that has

\textsuperscript{3}Mechanical Dynamics Inc.
4.4 Linear ADAMS model

Figure 4.7. The process of linearizing the ADAMS external tire model

The properties stiffness and damping in six degrees of freedom. The stiffness and damping factors are low in the horizontal plane and the rotation around the vertical axis, while the other rotational directions and the vertical plane have greater factors. If the velocity of the vehicle is constant, the damping in the horizontal plane can be chosen to resemble rolling resistance.

The tire forces acting on the vehicle have to be applied at the right point of action, and hence a link arm was created on each wheel corner, illustrated as (1) in Figure 4.8. The length of the link arm is the same as the radius of the tire, and the forces are applied at the lower end of it. The magnitude of the tire forces are added input signals to the generic assembly.

With these modifications the ADAMS external tires model is stable after the ADAMS/Linear’s linearization process. To complete the model it is connected to a tire model implemented in Simulink. This tire model has the same properties as the one used in the ADAMS model, explained in section 5.4. A simulation of the combined models, see block diagram in figure 4.9, displayed similar result as a co-simulation of the ADAMS model in figure 4.1.

To reach the last step in Figure 4.7 the Simulink command `linmod` is used on the combined models, see figure 4.10. The input and output signals are still steering angles and yaw rate. Unfortunately, even this linear model has positive eigenvalues, and is hence unstable. By increasing the low stiffness and damping factors in the bushings, a stable linear model is received. However, the result of such a simulation of the model is not satisfactory compared with the result of the ADAMS model.
4.4.2 Other modelling issues

Further tests were done in order to get around the problem with instability, and they are summarized in this section.

Removal of rigid body modes

When the lower stiffness and damping factors of the bushings are set to zero, a few and very small positive eigenvalues are received. A rigid body mode has an eigenvalue equal to zero which means that the stiffness and damping are zero, but numerical problem can make them vary a bit. In the linear ADAMS external tire model these modes are of no interest for the output signals and can therefore be removed. An example of how the removal can be done without affecting the system is illustrated in Appendix A. In the same way, the rigid body modes of the ADAMS external tires model are removed. A stable linear model is received, but linearizing the model together with the tire model still gives an unstable system.
4.4 Linear ADAMS model

Reduced amount of components

Another attempt to get a stable linear model is to reduce components of the ADAMS external tires model. The rotational spring-damper system and the flexible body are completely removed and replaced by rigid bodies. However, the result of the linearization is still an unstable model.

Simple model

A simple model is derived to serve as an example for the method of using an external tire model, see Figure 4.11. If this simple model is possible to linearize then, logically, it would be possible to linearize the ADAMS external tires model. The simple model consists of four bushings, with the stiffness and damping corresponding to the ADAMS external tires model, and a body mass. The tire forces, from the Simulink tire model, acts on the bushings in the same way as in the ADAMS external tires model. When co-simulating the simple ADAMS model it does not get the proper behavior, and thus it is not interesting to linearize. This attempt is not completely evaluated, and can probably result in a proper model for future work.
Figure 4.10. Block diagram used when linearizing from the input steering angles to the output $\dot{\Psi}$ in order to get a linear ADAMS model

Figure 4.11. The process of linearizing a simple ADAMS model

Dlinmod

In MATLAB/Simulink there are different possibilities to linearize models. The command `dlinmod` works well for the model derived in Chapter 5 and has also been used successfully in other projects using a co-simulation environment. This method, see Figure 4.12, has not been evaluated much but when tried it has not given a satisfactory result.
Figure 4.12. The process of linearizing the ADAMS model by dlinmod command
Chapter 5

Simulink model

In this chapter another vehicle model is derived. The model will have the possibility to set individual steering angles, in line with the concept of ACM, but in a more simple way compared with the ADAMS model in the previous chapter. The mathematical equations of the model are implemented and simulated in MATLAB/Simulink, see Section 3.2, and therefore the model is named the Simulink model. The reasons for developing the Simulink model are

1. to get understanding of the dynamic behavior in subsystems of the vehicle, such as suspension and tire forces.

2. if the model behaves similarly to the ADAMS model, then the models are most probably reliable. The comparisons between the models also ease the process of finding modelling errors.

3. to examine the possibility to use it in the development process of the control system.

5.1 Model overview

The Simulink model is a three dimensional vehicle with ten degrees of freedom moving with constant velocity. The model has four wheels where each wheel can be steered individually to represent the ACM system, see Chapter 2. Each wheel is connected to a suspension and a tire model. The suspension model is fairly simple, where one part of the vehicle’s mass is connected to a spring and a damper. The Simulink tire model is based on the same properties as the tires in the ADAMS model, and since the vehicle moves at constant velocity, it only experiences lateral forces\(^1\). The input signals to the model are individual steering angles. The structure of the implemented vehicle model is illustrated in Figure 5.1. The blocks in the figure represent some of the sections of this chapter.

\(^1\)The vehicle experiences no rolling resistance or air resistance e.t.c.
5.2 Coordinate systems

By the use of two coordinate systems, one earth fixed and one rotating and translating with the vehicle, it is possible to describe vectors in both global and relative coordinates. The latter is placed in the centre of gravity of the vehicle with the $x$-axis pointing forward, the $y$-axis pointing left and the $z$-axis pointing upwards, which is the ISO standard, [8]. The angles, in Figure 5.2, $\Phi$, $\Theta$ and $\Psi$ are called roll-, pitch- and yaw-angle respectively. The $X$, $Y$ and $Z$ coordinates are used to describe the position of the vehicle with respect to the global coordinate system. The relationship between the coordinate systems will be discussed in Section 5.7.

5.3 Forces and torques

The force and torque equilibrium equations are derived in the local coordinate system, the forces acting on the model are thus projected onto this coordinate system. Figure 5.3 shows forces on front and rear tire. As mentioned in the beginning of the chapter, only lateral forces are included in the tire model, i.e. no rolling resistance or other longitudinal forces are considered.

The sum of the projected forces $F_{x1}$, $F_{y1}$, $F_{z1}$ in the local coordinate system leads to the following equations

\[ \sum F_{x1} = 0, \quad \sum F_{y1} = 0, \quad \sum F_{z1} = 0. \]
5.3 Forces and torques

Figure 5.3. The forces acting on the tires are normal, \( F_N \), and lateral forces, \( F_c \). The latter appears due to the steering angle \( \delta \) of the wheel.

\[ F_{x1} = -F_c \sin(\delta_1) \cos(\Theta) - F_{N1} \sin(\Theta) \]
\[ F_{y1} = F_c \cos(\delta_1) \cos(\Phi) + F_{N1} \sin(\Phi) \]
\[ F_{z1} = F_{N1} \cos(\Theta) \cos(\Phi) - (-F_c \sin(\delta_1)) \sin(\Theta) - F_c \cos(\delta_1) \sin(\Phi) \]
\[ F_{x2} = -F_c \sin(\delta_2) \cos(\Theta) - F_{N2} \sin(\Theta) \]
\[ F_{y2} = F_c \cos(\delta_2) \cos(\Phi) + F_{N2} \sin(\Phi) \]
\[ F_{z2} = F_{N2} \cos(\Theta) \cos(\Phi) - (-F_c \sin(\delta_2)) \sin(\Theta) - F_c \cos(\delta_2) \sin(\Phi) \]
\[ F_{x3} = F_c \sin(\delta_3) \cos(\Theta) - F_{N3} \sin(\Theta) \]
\[ F_{y3} = F_c \cos(\delta_3) \cos(\Phi) + F_{N3} \sin(\Phi) \]
\[ F_{z3} = F_{N3} \cos(\Theta) \cos(\Phi) - F_c \sin(\delta_3) \sin(\Theta) - F_c \cos(\delta_3) \sin(\Phi) \]
\[ F_{x4} = F_c \sin(\delta_4) \cos(\Theta) - F_{N4} \sin(\Theta) \]
\[ F_{y4} = F_c \cos(\delta_4) \cos(\Phi) + F_{N4} \sin(\Phi) \]
\[ F_{z4} = F_{N4} \cos(\Theta) \cos(\Phi) - F_c \sin(\delta_4) \sin(\Theta) - F_c \cos(\delta_4) \sin(\Phi) \]

Since the vehicle is moving, the use of Newton’s second law of motion gives the following relations

\[ \sum F_x = F_{x1} + F_{x2} + F_{x3} + F_{x4} = m \dot{v}_x \]
\[ \sum F_y = F_{y1} + F_{y2} + F_{y3} + F_{y4} = m \dot{v}_y \]
\[ \sum F_z = F_{z1} + F_{z2} + F_{z3} + F_{z4} - mg = m \dot{v}_y \]
\[ J_\phi \ddot{\phi} = (F_{x1} + F_{x3})c - (F_{x2} + F_{x4})c + F_{y1}z_1 + F_{y2}z_2 + F_{y3}z_3 + F_{y4}z_4 \]
\[ J_\theta \ddot{\theta} = -(F_{x1} + F_{x2})a + (F_{x3} + F_{x4})b - F_{x1}z_1 - F_{x2}z_2 - F_{x3}z_3 - F_{x4}z_4 \]
\[ J_\psi \ddot{\psi} = (F_{x1} + F_{x4})c - (F_{x2} + F_{x3})c + (F_{y1} + F_{y2})a - (F_{y3} + F_{y4})b \]
The coordinates $z_i$ are calculated and approximated from the global coordinates $Z$ and the initial height, $h$, of the centre of gravity as follows

$$z_1 = h + (Z - a\Theta + c\Phi)$$
$$z_2 = h + (Z - a\Theta - c\Phi)$$
$$z_3 = h + (Z + b\Theta + c\Phi)$$
$$z_4 = h + (Z + b\Theta - c\Phi)$$

### 5.4 Tire model

If the Simulink model is to resemble the ADAMS model it would be preferable if they both use a tire model with similar properties. The ADAMS model uses a tire model based on the magic formula

$$F_y = F_{y0}(\alpha, \gamma, F_z)$$
$$F_{y0} = D_y \sin (C_y \arctan (B_y \alpha_y - E_y (B_y \alpha_y - \arctan (B_y \alpha_y))))$$
$$\alpha_y = \alpha + S_{Hy}$$

(5.1)

The coefficients in the formula are calculated from parameters measured on a physical tire, which makes the model realistic. The magic formula was developed by Pecejka, [8], and is the most common way of modelling tire forces. It makes it possible to determine lateral as well as longitudinal forces and aligning moment. However, using the magic formula for the combination of lateral and longitudinal forces make the formula complicated, but since the Simulink model only experiences lateral forces, it is simple to use. The Equation 5.1 is used only to determine the lateral force of a tire and is similar to the one used in the ADAMS model, [2]. The lateral force is a function of the variables slip angle, $\alpha$, the normal force acting on the tire, $F_z$, and the camber angle, $\gamma$. The coefficients in Equation 5.1 are determined by parameters from the ADAMS Tire property file, see Table 5.1.

<table>
<thead>
<tr>
<th>Stiffness factor</th>
<th>$B_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shape factor</td>
<td>$C_y$</td>
</tr>
<tr>
<td>Peak factor</td>
<td>$D_y$</td>
</tr>
<tr>
<td>Curvature factor</td>
<td>$E_y$</td>
</tr>
<tr>
<td>Horizontal shift</td>
<td>$S_{Hy}$</td>
</tr>
</tbody>
</table>

**Table 5.1.** The tire property that each coefficient affect

A common way when modelling tires is to neglect the influence of horizontal and vertical shift, [8]. Consequently, all the parameters used for calculating the shift, in Table 5.1, are set to zero. A tire model that includes vertical and or horizontal shift experiences a static lateral force. Ignoring the shifts will thus make the static force disappear. Since it is not possible to vary the camber angle in the Simulink
model it is also natural to ignore the influence from that variable. Removal of the camber parameter simplifies the formula even more. The lateral forces end up being functions of $\alpha$, see Figure 5.4. Notice that the coefficient $D_y$ varies due to the variation of the normal forces, see Section 5.5. The slip angle is calculated by

\[\alpha = \delta - \arctan \left( \frac{v_y}{v_x} \right)\]

The velocity on each tire is calculated from its relation to the velocity and rotation in the centre of gravity, see figure 5.5, by the following equations

\[
\begin{align*}
\alpha_1 &= \delta_1 - \arctan \left( \frac{v_{y1}}{v_{x1}} \right) \\
\alpha_2 &= \delta_2 - \arctan \left( \frac{v_{y2}}{v_{x2}} \right) \\
\alpha_3 &= -\delta_3 - \arctan \left( \frac{v_{y3}}{v_{x3}} \right) \\
\alpha_4 &= -\delta_4 - \arctan \left( \frac{v_{y4}}{v_{x4}} \right)
\end{align*}
\]
5.5 Suspension model

The ACM concept includes a sophisticated model of the suspension. Deriving the equations for that is not realistic for the Simulink model. Instead a simple one dimensional suspension model is used, which neglects the stiffness in the tires, see Figure 5.6. The spring and damper constants are chosen to correspond to the ADAMS model. The mass, $m$, acting on the suspension system of each wheel corner is calculated taking the location of centre of gravity into account. The total weight is the same as the weight in the ADAMS model. Force equilibrium between the mass and the wheel gives the following expression for the normal force:

$$m \ddot{z} = -k z - d \dot{z} - mg$$

$$0 = F_N + k z + b \dot{z} \Rightarrow F_N = -k z - b \dot{z}$$
5.6 Dynamics

The spring-damper system that is applied on each tire

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{spring_damper_system.png}
\caption{The spring-damper system that is applied on each tire}
\end{figure}

where \( z_i \) and \( \dot{z}_i \) are,

\[
\begin{align*}
  z_1 &= Z - a\Theta + c\Phi \\
  z_2 &= Z - a\Theta - c\Phi \\
  z_3 &= Z + b\Theta + c\Phi \\
  z_4 &= Z + b\Theta - c\Phi \\
  \dot{z}_1 &= Z - a\dot{\Theta} + c\dot{\Phi} \\
  \dot{z}_2 &= Z - a\dot{\Theta} - c\dot{\Phi} \\
  \dot{z}_3 &= Z + b\dot{\Theta} + c\dot{\Phi} \\
  \dot{z}_4 &= Z + b\dot{\Theta} - c\dot{\Phi}
\end{align*}
\]

and since the angles are relatively small \( \sin(\Phi, \Theta) \approx (\Phi, \Theta) \) is a reasonable approximation.

5.6 Dynamics

The following relationship is used for the time derivative of the vector \( V \), calculated relatively to a rotating coordinate system, [13].

\[
\left( \frac{dV}{dt} \right)_{XYZ} = \left( \frac{dV}{dt} \right)_{xyz} + \omega \times V
\]

Since the result of the force- and momentum equilibrium equations in Section 5.3 is the derivative of the global velocity vector, the formula is used to calculate the local velocity vector.

\[
\left( \frac{dV}{dt} \right)_{xyz} = \left( \frac{dV}{dt} \right)_{XYZ} - \omega \times V
\]
\[ \sum F = \dot{p} = m\dot{v} \]
\[ \sum M = \dot{H} = J\dot{\omega} \]

\[ \Rightarrow \]
\[ \sum F - \omega \times p = \dot{p} \]
\[ \sum M - \omega \times H = \dot{H} \] \hspace{1cm} (5.2)

Equations 5.2, give the correct velocity and rotation vector in the local vehicle coordinate system.

### 5.7 Global positioning

As illustrated in Figure 5.1 it is also necessary to know the vehicle’s global position.

\[
R = \begin{pmatrix}
\cos(\Psi) & -\sin(\Psi) & 0 \\
\sin(\Psi) & \cos(\Psi) & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\begin{pmatrix}
\cos(\Theta) & 0 & \sin(\Theta) \\
0 & 1 & 0 \\
-\sin(\Theta) & 0 & \cos(\Theta) \\
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 0 \\
0 & \cos(\Phi) & -\sin(\Phi) \\
0 & \sin(\Phi) & \cos(\Phi) \\
\end{pmatrix}
\]

\[
v_{XYZ} = R(\Phi, \Theta, \Psi)v_{xyz} \] \hspace{1cm} (5.3)

By the use of the rotation matrices, the local velocity vector is projected on the global axis system by Equation 5.3.

### 5.8 Linear Simulink model

Simulink provides the functions \texttt{linmod} and \texttt{dlinmod}, which extract linear models in the form of the state-space matrices \(A, B, C,\) and \(D\)

\[
\dot{x} = Ax + Bu \\
y = Cx + Du 
\]

where \(x, u,\) and \(y\) are state, input and output vectors. The model inputs and outputs must be defined, which enable several possibilities for studying different signals in a linear environment. The state-space model can be analyzed with all the functions and tools MATLAB provides.

In order to be certain that the system is not overmodelled, i.e. is a minimal realization, a function \texttt{minreal} can be used. A system that is a minimal realization is observable and controllable, [5]. An observable system only consists of the states necessary to determine the output signal, while an input signal in a system that is controllable can affect each state of the system.

Since the model derived in this chapter is in the continues time domain, the command \texttt{linmod} is used. In case of a discrete model, the command \texttt{dlinmod} is more appropriate. \(\delta_i\) are defined as input signals and \(\Psi\) as output signal in the non-linear model. The MATLAB command sequence to put a non-linear model, called \texttt{sim_model}, on a linear minimal realization state space form are
\[[A, B, C, D] = \text{linmod}(\text{sim\_model})\]
\[\text{lin\_sim\_model} = \text{ss}(A, B, C, D)\]
\[\text{lin\_sim\_model\_min} = \text{minreal}(\text{lin\_sim\_model})\]

The linear model before minreal has 15 states, while the linear model after minreal has 4 states\(^2\). The poles of the linear model are illustrated in Figure 5.7 and they are all located in the left half-plane, which means that the linear model is stable. The Bode diagram in Figure 5.8 shows the magnitude of the yaw rate and body slip for different frequencies.

\[\begin{array}{c}
\text{Figure 5.7. The poles of the linear Simulink model.}
\end{array}\]

\(^2\)Notice that the linear model from now on always refers to the system after minreal.
Figure 5.8. The bode diagram of the linear Simulink model. Input signals are front steering angles and output signals are body slip and yaw rate.
Chapter 6

Model evaluation

The ADAMS and Simulink models will be used to design and evaluate the latter control system. A comparison between the two models is made in order to detect modelling errors. This chapter also deals with the driving scenario used for evaluation of the ADAMS and Simulink model, and thus, also for the latter yaw rate control system.

6.1 Driving scenario

Since we are studying the possibilities for yaw control by applying the ACM-concept it is important to derive a relevant driving scenario. The controller will be restricted to small slip and steering angles and, therefore stability during a lane change on a more or less slippery surface will be of interest.

The time for a complete lane change maneuver and the speed of the vehicle determines the magnitude of the steering angle. The aim, in this case, is to set the circumstances so that the car travels at a velocity of 90 km/h and move approximately 3-4 m sidewise, similar to a normal lane-change maneuver. The friction coefficient, \( \mu \), will then be reduced to validate the controller to be derived in Chapter 7. The front steering angle, \( \delta_f \), in the chosen lane change maneuver is plotted in Figure 6.1 and results in the desired scenario.

Ackermann angles

When the vehicle turns, each wheel follows a specific circular path determined by the turning radius. With a constant steering signal the inner wheels follows a tighter circle than the outer wheels, [4]. Thus the steering signal must be greater on the inner wheel than on the outer. The difference between the steering angles is called the Ackermann angle, see Figure 6.2.

The angle can easily be calculated by the equations below. The intended steer-
Figure 6.1. Steering angle $\delta_f$ in the lane change maneuver

Figure 6.2. Ackermann angle of a vehicle is $\delta_1 - \delta_2$

The steering angle from the driver is

$$\delta_f = \frac{\delta_1 + \delta_2}{2}$$
If $\delta_f > 0$ then

$$
\delta_1 = \arctan \left( \frac{a + b}{r + c} \right)
$$

$$
\delta_2 = \arctan \left( \frac{a + b}{r - c} \right)
$$

$$
\delta_{ack} = \delta_1 - \delta_2
$$

where $r$ is the turning-radius and is given by

$$
r = \frac{a + b}{\tan(\delta_f)}
$$

when no slip is assumed. The coefficients $a$ and $b$ are the distance from center of gravity to the front respectively rear axle, and $c$ is the length of the wheel distance.

By using the same parameters in both the ADAMS and Simulink models with a maximum steering angle in the lane change, the influence of the Ackermann angle will be very small. Therefore it will be neglected in the simulations.

### 6.2 Tuning process

This section will describe the tuning of the ADAMS and Simulink model presented in Chapter 4 and 5 in order for them to show similar behaviour.

The tire model is one thing that differs, since the Simulink model does not use longitudinal forces, see Section 5.4. This limitation means that it is not possible to accelerate or decelerate the vehicle. In ADAMS, different tire models can be chosen. To make it similar to the Simulink model, a tire model without longitudinal forces is used. A common way of dealing with tires in vehicle models, is to neglect the horizontal- and vertical shift [8]. This is done here and the lateral forces in the models corresponds well.

As mentioned in Chapter 4, the ADAMS model is developed with forces applied on the cylinders, via a spring-damper system, as input signals instead of a steering angle. Dynamics in the ADAMS model make it hard to get an exact steering angle through open loop simulation. Therefore a PI-controller is implemented in ADAMS to get the desired steering angle.

Stiffness and damping factors in the rotational damper of the ADAMS model are increased to use similar values as in the Simulink model. The camber angle will influence the lateral forces, even though it is very small. It is included in the ADAMS model but neglected in the Simulink model. Studying the equations for the tire model, the camber angle will not have a major affect on the result for the lane change maneuver. However, in other driving scenarios this may be worth considering.
6.3 Non linear model comparison

Non-linear vehicle models should be compared in many different situations in the time domain in order for them to be considered to correspond to one another. In this section though, only the lane change described in Section 6.1 is studied. This is due to the circumstances under which the yaw rate controller is supposed to be used.

When developing a yaw rate controller the signals body-slip, \( \beta \), and of course yaw-rate, \( \Psi \), could be of interest, [8]. These signals are illustrated in Figure 6.3.

![Figure 6.3. Signals \( \beta \) and \( \Psi \) on a vehicle](image)

6.3.1 Speed and movements

The steering inputs are the same in both the ADAMS and Simulink model. The rear steering angles, \( \delta_r \), are kept close to zero and the front steering angles, \( \delta_f \), are illustrated in Figure 6.4.

As seen in Figure 6.4 the signal \( \beta \) differs between the models. One reason for this is due to the difference in complexity. The Simulink model does not include as much dynamic behavior of different parts in the vehicle as the ADAMS model. \( \beta \) is calculated by using the vehicle’s local velocities \( v_x \) and \( v_y \).

\[
\beta = \arctan \left( \frac{v_y}{v_x} \right) \tag{6.1}
\]

Since \( v_x \) stays very close to the initial velocity it does not make Equation 6.1 vary much. The problem is \( v_y \), see Figure 6.5. This means that the vehicle modelled in ADAMS does not take the same path in a lane change maneuver as the one modelled in Simulink. But as seen in the XY plot in Figure 6.4, the difference does not affect the result much. \( \Psi \) is of course an important signal for the yaw rate controller. As seen in Figure 6.4, the models show similar responses with respect to this signal.
6.3 Non linear model comparison

Figure 6.4. Upper left plot: The steering angle applied on the ADAMS model and Simulink model. Upper right plot: The path that the models travel in the lane change maneuver. Lower left plot: The yaw rate that the models experience in the lane-change. Lower right plot: The body slip that the models experience in the lane change.

6.3.2 Forces

It is important that forces acting upon the models show similar behavior. The vehicle can be seen as an interface where the forces are applied. If the response of the two vehicles is similar, then a correspondence between the models for this driving scenario can be assumed. Since longitudinal forces are neglected, the only forces of interest are lateral- and normal forces, see Figure 6.6.

The normal forces differ between the two models due to different weight distribution. In the Simulink model the weight is a point mass while in the ADAMS model the mass is divided in tank, passengers, luggage, wheels e.t.c. and placed in different locations of the vehicle and are thus not symmetrically distributed. This is the reason why it is a small difference between left and right normal forces in the ADAMS model. The lateral force, also displayed in Figure 6.6, is partly
determined by the normal force and partly by the slip angle. The slip angle differs slightly between the models, see Figure 6.7, and therefore the magnitude of the lateral forces, displayed in Figure 6.6, differs from one model to the other. Since the slip angle is determined by $\Psi$, $v_x$ and $v_y$, see Section 5.4, it is possible to say that even this depends on the difference in $v_y$ seen in Figure 6.5.

### 6.4 Linear model evaluation

The operating point of the linearization is 90 km/h driving straight ahead. The $\Psi$ and $\beta$ signals of the linear Simulink model are compared with the corresponding signal of the non-linear Simulink model in Figure 6.7. This comparison is done to verify that it is reasonable to design a controller on this linear model. Both signals have similar behavior and it is hard to see any differences at all between the two models. This result is good considering the control design in the following chapter. Since the non-linear Simulink model resembles the ADAMS model a controller working for the Simulink model will hopefully also work for the ADAMS model.

### 6.5 Conclusion

In the lane change driving scenario, described in Section 6.1, the ADAMS model and the Simulink model give roughly the same dynamic behavior. This is far from saying that the models corresponds to one another in other scenarios. Though, this
Figure 6.6. Upper plots: Lateral and normal force on each wheel of the ADAMS model. Lower plots: Lateral and normal force on each wheel of the Simulink model.

is a sign that the non-linear models, in some sense, are similar for this particular lane change. The linear Simulink model corresponded well with the non-linear Simulink model, which is promising for the control design.
Figure 6.7. Upper plots: Slip angle of the wheels in the ADAMS- and Simulink model. Lower plots: The $\dot{\Psi}$ and $\beta$ for the linear and non-linear Simulink model.
Chapter 7

Active yaw control

The concept of ACM opens the door to many new possibilities in the aspect of control systems. This chapter describes two different yaw rate controllers that have been developed during this project. Controlling the yaw rate means controlling the speed of rotation around the z axis of the vehicle’s coordinate system, see Section 5.2. The driver of a vehicle has learned to feel the car dynamics under the normal operating conditions. But in critical situations, like slippery roads or a flat tire or an unexpected strong side wind, he notices that he does not master the vehicle’s yaw motion and may react in a wrong way. A yaw controller would, in this case, have a stabilizing effect and the driver would hopefully not feel the change of condition.

The objective for a yaw rate controller is not only to keep the vehicle stable in abrupt braking maneuvers and high speed cornering, but also when the surface of the ground is slippery. The two controllers derived in this chapter are only working under the circumstances of small front steering and slip angles. Therefore it might be enough that the compensated yaw is generated by the use of the steering possibilities in the concept of ACM. More abrupt breaking and steering maneuvers are considered to be in a different driving mode, for example collision avoidance, and may be solved using individual wheel breaking. The solution of the controllers is based on the fact that the driver is in complete control of the front steering angles, thus the aim for the yaw controllers are to make the intended driving maneuver successful by using only the rear wheels steering possibility. If the controllers are able to stabilize the non-linear ADAMS model, when subjected to the lane change maneuver presented in Section 6.1, they are considered successful. A comparison between two driving situations is evaluated, one on a firm surface and one on a slippery surface.
7.1 Control strategy

The strategy is to develop two controllers. One is based on proportional feedback and the other is developed by the use of linear quadratic theory. Due to the problems with linearization of the ADAMS model, see Section 4.4, the controllers are designed using the linear Simulink model. Working with linear models makes it possible to analyze characteristics of the controlled systems in the frequency domain and gives simpler conditions in the time domain. The controlled linear system is expressed in terms of the error between actual and desired values. The front wheel steering angle, $\delta_f$, is considered as a known disturbance to the system. This idea is used in the paper [12]. As a consequence of this, the sensitivity function, see [6], is used for controller evaluation. If the front steering angle has minimal influence on the error signal then the controllers have satisfactory properties.

7.1.1 Error system

The control object is a system developed in order to determine the difference between the desired yaw rate and the actual yaw rate. To achieve this the Simulink model is linearized analogous to Section 5.8, the input signals are $\delta_f$ and $\delta_r$, output signal is $\dot{\Psi}$. $\dot{\Psi}$ is considered to be the actual yaw rate and the linear system is expressed on state space form

$$\begin{align*}
\dot{x} & = Ax + B \begin{pmatrix} \delta_f \\ \delta_r \end{pmatrix} \\
\dot{\Psi} & = Cx
\end{align*}$$

(7.1)

The matrices $A$, $B$ and $C$ are generated by the linearization function. The desired value of the yaw rate is also preferably given on a state space form, its transfer function is put in Equation 7.2. The value of the parameters $K_r$ and $\tau_r$ are determined so that the curve of Equation 7.2 is similar to the yaw rate curve illustrating a proper behavior of the vehicle in the lane change driving scenario.

$$\begin{align*}
\dot{\Psi}_{des} & = \frac{K_r}{1 + \tau_r \cdot s} \delta_f \\
K_r & = 7.9963 \\
\tau_r & = 0.1000
\end{align*}$$

(7.2)

The transfer function is easily transformed into standard state space form

$$\dot{\Psi}_{des} = -\frac{1}{\tau_r} \dot{\Psi}_{des} + \frac{K_r}{\tau_r} \delta_f$$

(7.3)

The error system is obtained by subtracting the system in Equation 7.1 from the system for desired values, Equation 7.3.

$$\begin{align*}
\dot{x} & = A_{error} \ x + B_{error} \begin{pmatrix} \delta_f \\ \delta_r \end{pmatrix} \\
\Delta \dot{\Psi} & = C_{error} \ x
\end{align*}$$
7.1.2 Signal separation

As mentioned in the beginning of the chapter, the only way the yaw rate controller is allowed to affect the system is by \( \delta_r \). In order to determine the system describing \( \delta_r \)'s influence on \( \Delta \dot{\Psi} \) it is separated from the system describing \( \delta_f \)'s influence on \( \Delta \dot{\Psi} \). The latter is treated as a known disturbance \( w \) to the system, see Figure 7.1.

\[
\Delta \dot{\Psi} = \frac{1}{1 + G_{sys}P} w - \frac{G_{sys}P}{1 + G_{sys}P} n = S w - T n \tag{7.4}
\]

In Equation 7.4 \( S \) determines how the disturbance \( w \) affects \( \dot{\Psi} \), and \( T \) determines how the measurement noise \( n \) affects the same signal. \( T \) is also an indicator on
how well a controller handles model errors with respect to instability. In control theory the $S$-function is called sensitivity function [6], and it is analyzed in many situations when developing control systems. In this case it is of importance since the characteristics of the disturbance $w$ is known. The relation between $S$ and $T$ is

$$S + T = 1$$

### 7.2 Proportional feedback

Proportional feedback makes it easy to implement a controller and analyzing the characteristics of the system. The rate of change of the yaw angle, $\dot{\Psi}$, is measured in many Volvo cars and therefore the control system is realistic to accomplish. The proportional feedback controller is designed by taking the output signal $\dot{\Psi}$ from the control subject, see Section 7.1.1, multiplying it with a suitable constant and using it as an input signal for the rear wheel steering angle, $\delta_r$. Different values of the constant, $P$, give the system different characteristics.

#### 7.2.1 P-controller characteristics

The sensitivity function, $S$, needs to be small for low frequencies, and based upon the Bode magnitude diagram of $G_{sys}$ in Figure 7.3, it is found that a proportional controller is enough to fulfill that criterion, see Equation 7.4. Figure 7.4 shows the

![Figure 7.3. Bode magnitude diagram for $G_{sys}$](image)

sensitivity function and the complementary sensitivity function for some different
values of the proportional feedback. All of the chosen constants for the controller give $S$ the desired appearance! It is also interesting to analyze $T$, called the complementary sensitivity function. The higher the constant is, the less tolerant the system is for measurement noise, $n$. However, it is more interesting to study $T$ with respect to sensitivity for the model error $\Delta G$. The condition for complementary sensitivity function $T$ are:

$$|T(i\omega)| < \frac{1}{\Delta G(i\omega)} \forall \omega$$  \hspace{1cm} (7.5)$$

Since the aim for the controller is to work on the non-linear ADAMS model, robustness against model errors is needed. $T$ is below 0 dB for all frequencies, see Figure 7.4, which according to Equation 7.5 means a more than 100 percent tolerance against model errors. Notice that this only refers to linear systems, but still it is here considered an approximate measure. The proportional controller is evaluated in Section 7.4 with the constant $P = 1.3$.

\[\text{Figure 7.4. Sensitivity and complementary sensitivity function for the proportional feedback controller}\]
7.3 Linear quadratic control

In this section a controller, $F_y$, is derived by the use of linear quadratic theory. It is a more sophisticated method for controlling the yaw rate than the proportional feedback described in the previous section. A different error system is needed and this system is called, the extended error system. Beside the $\Delta \dot{\Psi}$ signal it also includes the body slip error, $\Delta \beta$. The term body slip refers to the angle, in the centre of gravity, between the travel direction and the vehicle’s pointing direction. In [3] the yaw rate controller keeps $\beta \equiv 0$ by the use of steering the rear wheels. This is possible since the controller uses the front wheels to generate the desired yaw rate. In [12] $\beta$ is controlled close to zero by a $H_\infty$ controller. The controller distributes a breaking force to the wheels to generate the compensating yaw motion. Consequently, it is interesting to include $\beta$ in the yaw controller derived in this project to see how it affects the result. $\beta$ is normally not measurable on a vehicle and therefore a Kalman filter is implemented to solve the problem by estimating the signal. Mathematical relations lead to an optimal controller, based on the decisions made on weight matrices in the LQG-gain and design parameters in the Kalman filter, [6].

7.3.1 Extended error system

The $\beta$- and $\dot{\Psi}$-signals, are determined analogous with the error system in Section 7.1.1. The Simulink model is linearized with the input signals $\delta_f$, $\delta_r$, and the output signals $\beta$ and $\dot{\Psi}$. The desired value for body slip, $\beta_{des}$, is set to zero, [3] and [12]. The signal $\dot{\Psi}_{des}$ is derived the same way as for the proportional controller. The extended error system is obtained by the desired signals subtracted from the actual.

The intention is, also in this case, to minimize the effect of $\delta_f$ on the signals $\Delta \dot{\Psi}$ and $\Delta \beta$. Therefore is $\delta_f$ considered a known disturbance. Figure 7.5 illustrates the procedure used to separate the extended error system in four different subsystems. The separated system has two output signals that are affected by two different disturbance signals, $w_1$ and $w_2$, which originates from $\delta_f$.

$G_{sys1}$ and $G_{sys2}$ forms a SIMO\(^1\) system that is the foundation for the controller $F_y$. The controller is easier to derive by having the system expressed in state space form

$$\begin{align*}
\dot{x} &= A_{sys} x + B_{sys} \delta_r \\
y &= C_{sys} x
\end{align*}$$

7.3.2 LQG controller

The LQG controller $F_y$, see Figure 7.6 for structure, consist of a Kalman filter $K$ and a constant feedback $L$ that is decided by linear quadratic theory.

\(^1\)Single Input Multiple Output
7.3 Linear quadratic control

Figure 7.5. The extended error system separated into two systems with $\delta_r$ as input signal and a disturbance $\delta_f$. The output signals are $\Delta \beta$ and $\Delta \Psi$.

Figure 7.6. The LQG controller $F_y$

The Kalman filter is used to estimate the non measurable signal $\Delta \beta$. The system in Equation 7.6 is extended with state disturbances and measurement noise, analogous with [6]

$$
\begin{align*}
    \dot{x} &= A_{sys} x + B_{sys} \delta_r + Nw \\
    y &= C_{sys} x + n
\end{align*}
$$

(7.7)
where \( N = [1..1]^T \). The equation of the Kalman filter is

\[
\dot{\hat{x}} = A_{sys} \hat{x} + B_{sys} \delta_r + K(\Delta \dot{\Psi} - C \hat{x})
\]  (7.8)

where \( C \) equals the row of values in the matrix \( C_{sys} \) used for determining \( \Delta \dot{\Psi} \). The states \( \hat{x} \) are the estimated states in the Kalman filter. The difference between the estimated and measured signal \( \Delta \dot{\Psi} \) influence the states in the Kalman filter via the Kalman gain \( K \). In Figure 7.6 the equation is illustrated together with the LQG gain \( L \) in a block chart, which is the structure of the controller \( F_y \). \( K \) is determined by

\[
K = (P C^T + NR_{12}) R_2^{-1}
\]

where \( P \) is the symmetric positive semidefinite solution to the Ricatti equation

\[
AP + PA^T - (PC^T + NR_{12}) R_2^{-1} (PC^T + NR_{12})^T + NR_1 N^T = 0
\]

\( R_1 \) and \( R_2 \) are supposed to be the disturbance intensities of \( w \) and the measurement disturbance \( n \), but according to [6] they can be considered as design parameters to reach an acceptable estimation of \( \Delta \beta \). A rule of thumb is that greater \( R_1 \) than \( R_2 \) will suppress the sensitivity function, which is valuable for the controller in this project. It also resulted in a satisfying estimation of \( \Delta \beta \). The cross intensities, \( R_{12} \) and \( R_{21} \), are set to zero and the Riccati equation is solved by the use of the MATLAB command \( lqe \).

The constant \( L \) in Figure 7.6 is determined by optimizing the cost function

\[
J = \int (y^T Q_1 y + u^T Q_2 u) \, dt
\]  (7.9)

The equation is solved by the use of the MATLAB command \( lqry \) by choosing the parameters \( Q_1, Q_2 \) and set \( u = -L \hat{x} \). \( Q_1 \) was chosen greater than \( Q_2 \) to reach a solution that will minimize the output signal more then the input signal. The solution results in an optimal state feedback \( L \), [6].

### 7.3.3 LQG controller characteristics

The control objective for \( F_y \) is to keep the influence of \( w_1 \) and \( w_2 \) in Figure 7.5 as low as possible. However, since the system in Equation 7.6 is a SIMO system where only one of the outputs is fed back, it is more interesting to study the influence of \( \delta_f \) on the output signals. From Figure 7.7 two functions are derived that correspond to the sensitivity function described in Section 7.2.1. The main idea is to see how the signal \( \delta_f \) affects the output signals \( \Delta \beta \) and \( \Delta \dot{\Psi} \) with and without feedback. The uncontrolled system’s output signals are influenced by \( \delta_f \) through the following transfer functions.

\[
\begin{align*}
\Delta \beta &= G_{w1} \delta_f \\
\Delta \dot{\Psi} &= G_{w2} \delta_f
\end{align*}
\]  (7.10)
The controlled system’s output signal of the controlled system is influenced through the transfer functions $S_1$ and $S_2$ derived as follows,

\[
\Delta \beta = w_1 + G_{sys_1} \delta_r = G_{w_1} \delta_f + G_{sys_1} \delta_r \\
\delta_r = -F_y \Delta \Psi = -F_y (w_2 + G_{sys_2} \delta_r) = -F_y (G_{w_2} \delta_f + G_{sys_2} \delta_r) \\
\Rightarrow \delta_r = \frac{-F_y G_{w_2}}{1 + F_y G_{sys_2}} \delta_f \\
\Rightarrow \Delta \beta = \left( G_{w_1} - \frac{G_{sys_1} F_y G_{w_2}}{1 + F_y G_{sys_2}} \right) \delta_f = S_1 \delta_f \tag{7.11}
\]

\[
\Delta \Psi = w_2 + G_{sys_2} \delta_r = w_2 - G_{sys_2} F_y \Delta \Psi \\
\Rightarrow \Delta \Psi = \frac{1}{1 + G_{sys_2} F_y} w_2 = \frac{G_{w_2}}{1 + G_{sys_2} F_y} \delta_f = S_2 \delta_f \tag{7.12}
\]

Notice that $S_2$ corresponds to the sensitivity function for the extended system. The relation between the uncontrolled and the controlled system’s transfer functions in Equations 7.10-7.12 illustrates the improvement of the implementations of the controller. The equations are

\[
\frac{S_1}{G_{w_1}} \tag{7.13}
\]

\[
\frac{S_2}{G_{w_2}} \tag{7.14}
\]

and they are plotted in Figure 7.8. The influence of $\delta_f$ on the signal $\Delta \Psi$ with the LQG controller is small for low frequencies, see Figure 7.8, which is the objective with the controller. Unfortunately the $\Delta \beta$ signal, illustrated in the same figure,
Figure 7.8. The diagrams illustrate the Bode magnitude of the systems in Equations 7.10-7.12. They state the improvement of $\Delta \beta$ and $\Delta \dot{\Psi}$ between a LQG controlled system and an uncontrolled system. For the frequencies that the graphs are suppressed below one, the feedback improves the system.

does not show similar behavior. The controller can not affect this signal in a way that make the influence of $\delta_f$ much smaller. However $\Delta \beta$ is small compared with $\Delta \dot{\Psi}$, which indicates that the system has proper behavior anyway.

### 7.4 Evaluation of the controllers

The sensitivity and complementary sensitivity function for the proportional feedback and the LQG controller, see Figures 7.9 and 7.10, show similar behavior. The LQG controller has slightly lower sensitivity function for lower frequencies while the disturbance strengthens for higher frequencies. However, the peak will not affect the system since the disturbance frequencies are lower than 10 Hz. The plots indicate that the proportional feedback and LQG controller will show similar result in a simulation.

Since the $\Delta \beta$ signal is not affected so much by the LQG controller, see Figure 7.8, it will behave the same way when using a proportional feedback.

The evaluation of the controllers in this chapter consists of tests on three different models, the linear Simulink model, the non-linear Simulink model and the ADAMS model. Each model experiences four types of lane change.
7.4 Evaluation of the controllers

Figure 7.9. Sensitivity function for P- and LQG controller, transfer function from $\delta_f$ to $\Delta \Psi$

1. Without controller and with friction coefficient $\mu = 0.7$
2. Without controller and with $\mu = 0.07$
3. With the proportional feedback and with $\mu = 0.07$
4. With the LQG-controller and with $\mu = 0.07$

The values $\mu = 0.7$ corresponds to dry asphalt and $\mu = 0.07$ corresponds to ice.

7.4.1 Evaluation using Simulink model simulations

Simulation of the linear Simulink model gives a similar result in the first three lane changes. The difference in the lane change maneuver between $\mu=0.7$ and $\mu=0.07$ is small for the signal $\Psi$, which is due to the linearization of the tire model. A variation of the friction coefficient does not affect the linear model much. Recall the Figure 5.4, a linearization under the circumstances of small slip angles give a linear model the form of a straight steep graph. This graph is more or less steep depending on the friction factor $\mu$. However, with small slip angles the lateral forces are almost the same for both $\mu = 0.7$ and $\mu = 0.07$. Since the $\Psi$ signal in the lane change maneuvers does not differ, there is nothing to control for the linear Simulink model, see Figure 7.11. Instead, the focus will be on controlling the non-linear Simulink model and the ADAMS model.
The non-linear Simulink model shows a significant decrease of the $\dot{\Psi}$ value between the first two lane changes, see Figure 7.12. Using the proportional feedback stabilizes the vehicle satisfactory and, as expected. The LQG controller shows similar result. The difference between desired $\dot{\Psi}_{des}$ and actual $\dot{\Psi}$ with a proportional feedback and LQG controller respectively, is illustrated in Figure 7.13. Again, it is hard to tell the difference between the results of the controlled systems due to their similar behavior. Although, both controllers decrease the $\Delta\dot{\Psi}$ significantly. The rear steering angles, needed to generate the compensating yaw motion, are relatively small during the lane change for both control systems, see Figure 7.14. The XY plot for the lane change maneuvers is seen in Figure 7.15. Both controlled systems show similar behavior.

### 7.4.2 Evaluation using ADAMS model simulations

The complementary sensitivity function, see Figure 7.10, has a margin for model errors using the controllers. Therefore the controllers will hopefully work just as well for the ADAMS model as for the Simulink model. Figure 7.16 show that the LQG controller stabilizes the ADAMS model almost as good as the Simulink model, as expected. The $\Delta\Psi$ signals differs between the Simulink and ADAMS model, see Figures 7.13 and 7.17. First of all, it is a difference between the proportional feedback and the LQG controller for the ADAMS model, which was not found
when controlling the Simulink model. Secondly, the magnitude of the $\Delta \dot{\Psi}$ has increased for the ADAMS model, compared with $\Delta \dot{\Psi}$ in the Simulink model. These differences are based on the fact that the ADAMS model is more complex and therefore shows a more non-linear behavior. Although, both control systems fulfill the objective to reduce the $\Delta \dot{\Psi}$ error. The rear steering angles, needed to generate the compensating yaw motion in the ADAMS model, are relatively large compared to the ones needed for the Simulink model, see Figures 7.18 and 7.14. Notice that their magnitudes also are larger than the front steering angle, see Figure 6.1. However, the proportional feedback has smaller rear steering angles than the LQG controller, yet larger than the front steering angle. Since the $\Delta \dot{\Psi}$ was satisfactory reduced, these steering angles are necessary. The XY plots for the lane change maneuvers are seen in Figure 7.19. Both controlled systems show similar behavior.

7.4.3 Conclusion

The XY plots of the non-linear Simulink model and ADAMS model are interesting, see Figures 7.15 and 7.19. Let’s call the path that the vehicle travels when $\mu = 0.7$ for the reference path. Comparing the reference path with the paths of the two non-linear vehicles on slippery surface, with or without controller, shows a difference between the two models. In the uncontrolled lane change where $\mu = 0.07$, the ADAMS model has problem following the reference path. Figures 7.16 and 7.21 illustrates that the ADAMS model is well below the $\dot{\Psi}_{des}$ and has an increased $\beta$ signal, i.e. it is travelling out of the reference path not only due to decrease and delay of rotation, but also sliding. The Simulink model in the same lane change scenario also has lower rotation than the $\dot{\Psi}_{des}$ signal, see Figure 7.12, but compared with the ADAMS model it has lower body slip $\beta$, see Figure 7.20. The lane change for the Simulink model is illustrated in Figure 7.15, the difference between the Simulink and ADAMS model is obvious studying how far the vehicles travels in the Y-direction and how far from the reference path it is, for the ADAMS graph see Figure 7.19. This is explained by the delay of rotation in the ADAMS model, see Figure 7.16, and magnitude of the body slip signal $\beta$, see Figures 7.20 and 7.21. These differences are based on factors due to that the models are derived differently. Since the LQG controller and the proportional feedback shows similar results for the lane change, the rest of this section is only going to concern the LQG controller.

The non-linear Simulink model is stabilized with the LQG controller and follows the reference path much better than the uncontrolled model, see Figure 7.15. The ADAMS model applied with the LQG controller actually shows worse result, in the XY-plot, than the uncontrolled model, see Figure 7.19, even though the $\Delta \dot{\Psi}$ is smaller, see Figure 7.12. A comparison between the $\beta$ signal of the two non-linear models show the answer to this problem, see Figures 7.20 and 7.21. The ADAMS model has much more body slip than the Simulink model, in fact the controlled Simulink model has equal body slip with the uncontrolled ADAMS model. This observation leads to a conclusion, the controller has to pay more attention to the body slip, $\beta$, than the LQG controller in the Section 7.3 does. The influence that
the controller has on the $\beta$ signal is illustrated in Figure 7.8. One idea to solve this problem is described in paper [3], where the yaw controller uses both front and rear wheels to generate a compensating momentum and keep the body slip $\beta \equiv 0$, by controlling acceleration on the front axle, $\dot{\Psi}$ and $\beta$.

![Simulink linear $d\dot{\Psi}/dt$](image)

**Figure 7.11.** $\dot{\Psi}$ for the linear Simulink model without feedback and friction coefficient $\mu = 0.7$ and $\mu = 0.07$, and proportional feedback with $\mu = 0.07$
Figure 7.12. $\dot{\Psi}$ for the non-linear Simulink model without feedback and friction coefficient $\mu = 0.7$ and $\mu = 0.07$, and LQG feedback with $\mu = 0.07$

Figure 7.13. $\Delta \dot{\Psi}$ for the non-linear Simulink model without feedback, and with proportional and LQG feedback
Figure 7.14. $\delta_r$ for the non-linear Simulink model without feedback, and with proportional and LQG feedback

Figure 7.15. XY plot for the non-linear Simulink model without feedback and friction coefficient $\mu = 0.7$ and $\mu = 0.07$, and LQG feedback with $\mu = 0.07$
Figure 7.16. $\dot{\Psi}$ for the ADAMS model without feedback and friction coefficient $\mu = 0.7$ and $\mu = 0.07$, and LQG feedback with $\mu = 0.07$.

Figure 7.17. $\Delta \dot{\Psi}$ for the ADAMS model without feedback, and with proportional and LQG feedback.
Figure 7.18. $\delta_r$ for the ADAMS model without feedback, and with proportional and LQG feedback

Figure 7.19. XY plot for the ADAMS model without feedback and friction coefficient $\mu = 0.7$ and $\mu = 0.07$, and LQG feedback with $\mu = 0.07$
Figure 7.20. $\beta$ for the non-linear Simulink model without feedback and friction coefficient $\mu = 0.7$ and $\mu = 0.07$, and LQG feedback with $\mu = 0.07$.

Figure 7.21. $\beta$ for the ADAMS model without feedback and friction coefficient $\mu = 0.7$ and $\mu = 0.07$, and LQG feedback with $\mu = 0.07$.
Chapter 8

Conclusion

The conclusions of this project are divided into the parts Autonomous Corner Module, vehicle models, and control design. Each part also includes remarks on recommendations for future work.

8.1 Autonomous Corner Module

The concept of the Autonomous Corner Module enables many possibilities, not only in the aspect of safety and environment, but also additional features to make the vehicle more appealing. The steering actuators make it possible to set both camber and steering angle. However, the focus in this project has been on stabilization of the vehicle using the four wheel steering and therefore the possibility to control the camber angle has been neglected. Other interesting control systems for the future, in line with the concept of ACM, are an active suspension system and side wind and road slope compensating systems. The combination of these control systems are also worth to study, design and evaluate.

8.2 The ADAMS and Simulink model

Due to an acceptable behavior by the ADAMS model in a co-simulation environment, the first two problems from Section 1.3 are considered solved. An acceptable behavior is based on the fact that the signals in the ADAMS model resemble the model implemented in Simulink. Further improvement that can be done on the ADAMS model is to remove unnecessary details on the upper arm. These details make the model more complex than needed, which can lead to numerical problems when working with the model. Another component that also can be adjusted is the flexible body. Today it is not pre-tensioned, which forces the rotational spring-damper system to carry more of the load than it is supposed to.

In a comparison between the ADAMS/Car and the MATLAB/Simulink environment ADAMS/Car appears to be superior when the physical systems are com-
plex. ADAMS/Car is therefore definitely preferable to MATLAB/Simulink when modeling an advanced system like the Autonomous Corner Module. ADAMS/Car also provides some features that can not be found in MATLAB/Simulink, e.g. animation of the model and mode analysis.

The model implemented in Simulink is provided with a very simple wheel corner to resemble the concept of ACM. In a lane-change both the complex ADAMS model and the simple Simulink model shows similar behavior. A more accurate comparison between the models is done in the frequency domain, which requires linear models. In such a comparison it becomes obvious which dynamic behavior that is neglected in the simplification. Linearizing the Simulink model is no problem meanwhile the linear ADAMS model turns out to be unstable. By analyzing the model, and referring to a discussion with MDI, it is possible to say that the linearization problem lies in the tires of the ADAMS model. The linearization process is based on the fact that a force in the model rises due to stiffness or damping in different components. However, the tire forces do not depend on stiffness or damping. This fact leads to that the linear ADAMS model is not reliable. Some work-around to the problem is presented in Chapter 4, but unfortunately without a successful result. Future works, that can be done in order to hopefully avoid the unstable linear ADAMS model, are summarized in the following items.

- Use a different tire model than the one used for the Simulink model in Section 5.4, together with the ADAMS model with external tires, see Section 4.4.1.
- Develop a simple ADAMS model based on four bushings, see Section 4.4.2, with external tires implemented in Simulink. If this model can be linearized then, logically, a more complex model should also be possible to linearize.
- Remove all low frequency modes of the linear ADAMS model with external tires before linearizing it with the external Simulink tires.
- ADAMS/Car version 10 is used in this work. A latter version of the program may have developed the linearization function.

### 8.3 Control design

The linearization result of the ADAMS model affected the structure of the project, see Section 1.4, and therefore the two methods of developing the control system are not possible to compare. This linearization problem makes it not possible to design a control system based on an ADAMS/Car complete vehicle model. The linearization problem and the fact that the Simulink model corresponds with the ADAMS model, results in that the linear Simulink model is used in the control design.

In order to control the yaw rate, which is the last part of the problems in Section 1.3, controller based on proportional feedback and LQ feedback are developed. Both controllers show similar results on the two models. It is satisfactory on the
Simulink but worse on the ADAMS model. This difference in result probably depends on the complexity between the two models. As it turns out, it is important that not only the yaw rate follows the desired value but also that the body slip is kept close to zero at all time to get the proper behavior of the vehicle. The objective in this report is to use the rear wheel steering angles to generate the compensating yaw torque, as a consequence of this, the better the yaw rate get the worse the body slip get, which is obvious when the controllers are applied to the ADAMS model. An interesting idea, to get around this problem, is evaluated in the paper [3] where the front steering generates the yaw torque and the rear steering keeps the body slip equal to zero. The control variables are yaw rate, body slip and acceleration of the front axle.

The last conclusion is that the rear steering angles, which is set to control the yaw rate of the vehicle, needs good precision in order to achieve the desired steering angles. The maximum rear steering angle, in the lane-changes in this report, is 0.3°.
Bibliography


Appendix A

Two mass system

In this appendix the possibility to remove the rigid body modes in a controlled system is illustrated. The idea is to control the system in Figure A.1 for a constant value of the distance between \(x_2\) and \(x_1\) by the use of the force \(F\). Deriving a state space model by the use of Newton’s second law, results in two rigid body modes, i.e. velocity for each of the masses. The poles will be zero or very close to zero. The control objective of the system does not need the possibility to describe the velocities of the masses. Combining the MATLAB commands \textit{balmr} and \textit{modred} makes it possible to remove the modes without affecting the result of the control system.

\[
\ddot{x}_1 = \frac{1}{m_1} (-kx_1 + kx_2 - c\dot{x}_1 + c\dot{x}_2)
\]

\[
\ddot{x}_2 = \frac{1}{m_2} (kx_1 - kx_2 + c\dot{x}_1 - c\dot{x}_2) + \frac{F}{m_2}
\]
\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_1 \\
\dot{x}_2 
\end{pmatrix}
= 
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
-\frac{k_1}{m_1} & \frac{k_2}{m_1} & -\frac{c_1}{m_1} & \frac{c_2}{m_1} \\
\frac{k_1}{m_1} & -\frac{k_2}{m_1} & \frac{c_1}{m_1} & -\frac{c_2}{m_1}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_1 \\
x_2 
\end{pmatrix}
+ 
\begin{pmatrix}
0 \\
0 \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
F \\
m_2
\end{pmatrix}
\]

(A.1)

Apply the MATLAB command \(\text{[sysbm, tot, h]}=\text{balmr}(\text{sys}, 2, 0.01)\) to the system in equation A.1, called \(\text{sys}\), and it will end up in a balanced state space model. The new system was reduced by removing the states \(x_3\) and \(x_4\) with the command \(\text{rsys=modred(sysbm, 3:4, ’del’)}\) from MATLAB.
Sammanfattning
Abstract
The purpose of this project is to continue the work on a vehicle model developed in ADAMS/Car and applied with the concept of ACM (Autonomous Corner Module). The project is divided up in two parts. The objective of the first part is to setup a co-simulation environment between ADAMS/Car and MATLAB/Simulink, and evaluate the vehicle model. In the second part a yaw controller is developed using only the rear wheel steering possibilities. The controller will be evaluated when it is applied on the vehicle model. The approach is to develop two models, one simpler in MATLAB/Simulink and one more complex in ADAMS/Car, and verify that they show similar behavior. The models will then be linearized and the control design will be based on the most appropriate linear model. Most of the work has been developing and evaluating the two vehicle models in ADAMS/Car and MATLAB/Simulink. The result was a working co-simulation environment where an evaluation of two different controllers was made. Due to linearization of the ADAMS model was nsuccessful, the controllers were based on the simpler linear Simulink model. Both controllers show similar results. Tests on the ADAMS model showed that it is hard to control both the yaw rate and body slip only by rear wheel steering.
På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtidiga ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovanbeskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© [Daniel Westbom, Petter Frejinger]